



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - K141502

DETEKSI AREA HIJAU PADA DAERAH URBAN BERBASIS METODE EKSTRAKSI ORIENTED FAST AND ROTATED BRIEF (ORB) DARI CITRA UDARA DENGAN RESOLUSI SANGAT TINGGI

AZIS ARIJAYA
NRP 5112 100 155

Dosen Pembimbing
Dr. Darlis Herumurti, S.Kom, M.Kom
Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - K141502

DETEKSI AREA HIJAU PADA DAERAH URBAN BERBASIS METODE EKSTRAKSI ORIENTED FAST AND ROTATED BRIEF (ORB) DARI CITRA UDARA DENGAN RESOLUSI SANGAT TINGGI

AZIS ARIJAYA
NRP 5112 100 155

Dosen Pembimbing
Dr. Darlis Herumurti, S.Kom, M.Kom
Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - K141502

GREEN AREA DETECTION IN URBAN AREA BASED ON ORIENTED FAST AND ROTATED BRIEF (ORB) METHOD FROM VERY HIGH RESOLUTION AERIAL IMAGE

**AZIS ARIJAYA
NRP 5112 100 155**

**Supervisor
Dr. Darlis Herumurti, S.Kom, M.Kom
Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.**

**INFORMATICS DEPARTMENT
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

DETEKSI AREA HIJAU PADA DAERAH URBAN BERBASIS METODE EKSTRAKSI ORIENTED FAST AND ROTATED BRIEF (ORB) DARI CITRA UDARA DENGAN RESOLUSI SANGAT TINGGI

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Cerdas dan Visi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

AZIS ARIJAYA

NRP : 5112 100 155

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Dr. Darlis Herumurti, S.Kom, M.Kom
NIP: 19771217 200312 001 (pembimbing 1)

Prof. Dr. Ir. Joko Lianto Buliali, M.Sc
NIP: 19670727 199203 1 002 (pembimbing 2)

**SURABAYA
JUNI 2016**

[Halaman ini sengaja dikosongkan]

DETEKSI AREA HIJAU PADA DAERAH URBAN BERBASIS METODE EKSTRAKSI ORIENTED FAST AND ROTATED BRIEF (ORB) DARI CITRA UDARA DENGAN RESOLUSI SANGAT TINGGI

Nama Mahasiswa : Azis Arijaya
NRP : 5112 100 155
Jurusan : Teknik Informatika FTIf-ITS
Dosen Pembimbing 1 : Dr. Darlis Herumurti, S.Kom, M.Kom
Dosen Pembimbing 2 : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.

ABSTRAK

Pendeteksian Area tanaman hijau memiliki peranan penting untuk dapat melakukan perencanaan tata kota. Hal ini sangat berkaitan dengan tingkat pencemaran udara yang dapat mempengaruhi penduduk dan lingkungan sekitar, terutama pada kesehatan anak-anak. Zat-zat beracun yang dihasilkan dari pencemaran udara adapt mengurangi kemampuan berfikir anak, tingkat IQ menjadi rendah dan pertumbuhan fisik akan terganggu. Serta bisa mengganggu kesehatan kulit, yaitu menjadi kusam, penuaan dini, dan yang lebih parah adalah resiko penyakit kanker kulit. Hal ini yang menjadi tantangan dalam pendeteksian area tanaman hijau pada daerah urban.

Menggunakan Teknik tradisional untuk mendeteksi area hijau tanaman dianggap sudah ketinggalan jaman, karena memakan banyak biaya dan waktu. Oleh karena itu, diperlukan sebuah system deteksi area tanaman hijau yang menggunakan citra udara Red-Green-Blue (RGB) yang nantinya diubah menjadi Hue-Saturation-Value (HSV) untuk mencari koordinat dengan mempertimbangkan spectrum warna. Setelah itu dilakukan deteksi untuk membedakan area hijau tanaman dengan yang bukan tanaman, metode yang dilakukan adalah Oriented FAST and Rotated BRIEF (ORB). Metode thresholding HSV

dapat menghasilkan nilai rata-rata recall, precision dan F1 masing-masing sebesar 5.11%, 79.30%, dan 9.45%. Metode Ekstraksi Feature ORB menghasilkan rata-rata nilai recall, precision, dan F1 masing-masing bernilai 41.15%, 49.63% dan 37.62%.

Kata kunci: deteksi area hijau, citra udara, ekstraksi fitur, ORB

GREEN AREA DETECTION IN URBAN AREA BASED ON ORIENTED FAST AND ROTATED BRIEF METHOD FROM VERY HIGH RESOLUTION AERIAL IMAGE

Name : Azis Arijaya
NRP : 5112 100 155
Major : Teknik Informatika FTIf-ITS
Supervisor I : Dr. Darlis Herumurti, S.Kom, M.Kom
Supervisor II : Prof. Dr. Ir. Joko Lianto Buliali, M.Sc.

ABSTRACT

Area Detection importance of a green plants have a role for effective urban planning. It is strongly related to levels of air pollution can affect people and the environment, especially on the health of children. Toxic substances resulting from air pollution adapt reducing child's ability to think, to be low-level IQ and physical growth will be impaired. And It can damage the health of the skin, which becomes dull, aging, and the more severe is the risk of skin cancer. This is a challenge in the area detection of greenery in urban areas.

The using the traditional technique to detect areas of green plants is considered outdated, because it takes a lot of cost and time. Therefore, we need a system of area detection of greenery that use aerial imagery of the Red-Green-Blue (RGB), which will be converted into Hue-Saturation-Value (HSV) to figure takes into account the coordinates of the color spectrum. Once that was done to distinguish the detection of green area with plants that are not plants, is the method performed Oriented FAST and Rotated BRIEF (ORB). HSV thresholding method produces an average of recall, precision, and F1 of respectively 5.11%, 79.30%, and 9.45%. Extraction Feature ORB method produces an average of recall, precision, and F1 of respectively 41.15%, 49.63% dan 37.62%.

Keywords: green area detection, aerial image, feature extraction, ORB

DAFTAR ISI

LEMBAR PENGESAHAN.....	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan.....	3
1.3. Manfaat.....	3
1.4. Rumusan Permasalahan.....	4
1.5. Batasan Permasalahan	4
1.6. Metodologi	4
1.7. Sistematika Penulisan.....	6
BAB II DASAR TEORI.....	9
2.1. Citra Udara	9
2.2. HSV Image	10
2.3. Oriented FAST and Rotated BRIEF.....	11
2.3.1. FAST Orientation	13
2.3.2. Rotated BRIEF	15
2.4. Morfologi Citra.....	15
2.4.1. <i>Structuring Element</i>	15
2.4.2. Dilasi	16
2.4.3. Erosi	16
2.4.4. <i>Opening</i>	17
2.4.5. <i>Closing</i>	17
2.5. Perhitungan Kinerja Aplikasi	19
BAB III ANALISIS DAN PERANCANGAN SISTEM.....	21
3.1. Desain Umum Sistem	21
3.2. Data	23

3.2.1.	Data Masukan	23
3.2.2.	Data Proses	24
3.2.3.	Data Keluaran	25
3.3.	Perancangan Proses	25
3.3.1.	Konversi HSV.....	26
3.3.2.	Ekstraksi Fitur ORB	29
3.3.3.	Morfologi.....	30
3.4.	Perancangan Antarmuka Perangkat Lunak.....	32
BAB IV	IMPLEMENTASI	35
4.1.	Lingkungan Implementasi	35
4.2.	Implementasi Proses	35
4.2.1.	Implementasi Tahap Thresholding HSV	35
4.2.2.	Implementasi Tahap ekstraksi fitur ORB	37
4.2.3.	Implementasi Tahap Morfologi	38
4.2.4.	Implementasi Tahap Pencocokan Fitur	38
4.3.	Impelementasi Antarmuka Perangkat Lunak.....	39
BAB V	PENGUJIAN DAN EVALUASI	43
5.1.	Lingkungan Pengujian.....	43
5.2.	Data Uji Coba	43
5.3.	Skenario Uji Coba	44
5.4.	Skenario Pengujian 1: Perhitungan nilai recall dan precision menggunakan HSV channel.....	44
5.5.	Skenario Pengujian 2: Perhitungan nilai recall dan presicion pada proses ekstraksi <i>feature</i> ORB	46
5.6.	Evaluasi	48
BAB VI	KESIMPULAN DAN SARAN	49
6.1.	Kesimpulan.....	49
6.2.	Saran	49
DAFTAR PUSTAKA.....		51
Lampiran.....		53
BIODATA PENULIS.....		57

DAFTAR GAMBAR

Gambar 2.1 Citra udara Kota Sapporo, Jepang	9
Gambar 2.2 Contoh model HSV	11
Gambar 2.3 Diagram alir ORB.....	12
Gambar 2.4 Deteksi keypoint menggunakan FAST	13
Gambar 2.5. Contoh Operasi Dilasi	18
Gambar 2.6. Contoh Operasi Erosi	18
Gambar 2.7. Contoh Operasi Opening	18
Gambar 2.8. Contoh Operasi Closing.....	19
Gambar 3.1 Diagram alur Proses program.....	22
Gambar 3.2 Contoh citra udara masukan	23
Gambar 3.3 Contoh citra udara hasil deteksi.....	25
Gambar 3.4 Alur proses konversi HSV	26
Gambar 3.5 Contoh citra udara konversi HSV.....	27
Gambar 3.6 Contoh citra udara hasil thresholding HSV	28
Gambar 3.7 Contoh citra hasil thresholding HSV yang diperjelas	28
Gambar 3.8 Alur proses ORB	29
Gambar 3.9 Contoh citra udara hasil ekstraksi fitur ORB.....	30
Gambar 3.10 Contoh citra udara hasil morpholgi	31
Gambar 3.11 Contoh citra udara setelah ditandai area hijau.....	31
Gambar 3.12 Rancangan Halaman Antarmuka Perangkat Lunak	33
Gambar 4.1 Visual Indeks pada subplot.....	40
Gambar 4.2 Hasil Implemtasi Halaman Output	40
Gambar 5.1 Gambar Uji Coba.....	44
Gambar 5.2 Contoh Citra Hasil Penandaan Manual	45
Gambar 5.3 Citra Hasil HSV Channel	46
Gambar 5.4 Hasil Citra Ekstraksi Feature ORB.....	47

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 3.1 Data Proses	24
Tabel 3.2. Nama dan Fungsi Tombol	32
Tabel 4.1 Lingkungan Implementasi Sistem	35
Tabel 4.2 Nama dan fungsi tombol tampilan output	41
Tabel 5.1 Spesifikasi lingkungan uji coba.....	43

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Kode program untuk thresholding Gambar HSV.....	36
Kode Sumber 4.2 Kode Program mencari Keypoint dengan ORB	37
Kode Sumber 4.3 Kode program untuk morfologi gambar	38
Kode Sumber 4.4 Kode program untuk mencocokkan fitur.....	38
Kode Sumber 4.5 Kode Program untuk menampilkan halaman Output.....	39

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

1.1. Latar Belakang

Kawasan perkotaan (urban) adalah wilayah yang mempunyai kegiatan utama bukan pertanian dengan susunan fungsi kawasan sebagai tempat permukiman perkotaan, pemusatan dan distribusi pelayanan jasa pemerintahan, pelayanan sosial, dan kegiatan ekonomi. Kawasan urban cenderung mengalami permasalahan dengan tingginya tingkat pertumbuhan kendaraan bermotor yang mengakibatkan polusi udara yang semakin banyak. Serta peningkatan daerah industri yang menghasilkan asap cerobong pabrik yang akan membebaskan CO₂ dan CO ke udara [1]. Berdasarkan penelitian tentang tingkat polusi paling mematikan di suatu Negara berdasarkan laporan Bloomberg bahwa Indonesia menduduki peringkat delapan setelah China, India, Pakistan, Bangladesh, Nigeria, Rusia, dan Amerika Serikat [2].

Dampak yang ditimbulkan dari pencemaran udara tersebut mempengaruhi penduduk dan lingkungan sekitar, membahayakan kesehatan seluruh komponen biotik penyusun ekosistem, mengganggu keindahan dan kenyamanan lingkungan, terutama pada kesehatan anak-anak. Kandungan timbal yang berlebih dapat menghambat produksi sel-sel darah merah dan akan menimbulkan anemia. Selain itu zat-zat beracun hasil dari pencemaran udara dapat mengurangi kemampuan berfikir anak, tingkat *Intelligence quotient* atau yang biasa disebut dengan IQ menjadi rendah, dan pertumbuhan fisik yang terganggu. Serta bisa mengganggu kesehatan kulit menjadi kusam, penuaan dini, dan yang lebih parah adalah resiko penyakit kanker kulit.

Pencemaran udara juga dapat menyebabkan kebocoran lapisan ozon sehingga membuat keseimbangan ekosistem menjadi terganggu akibat efek rumah kaca. Asap kebakaran hutan dapat menyebabkan gangguan iritasi dan infeksi saluran pernapasan akut (ISPA). Pencemaran udara dapat menyebabkan terjadinya keracunan akibat pengikatan CO₂ hasil dari pencemaran udara, serta dapat menurunkan kualitas udara untuk pernafasan semua organisme, terutama manusia sehingga akan menurunkan derajat kesehatan masyarakat.

Pencemaran udara juga dapat menyebabkan hujan yang mempengaruhi kualitas tanaman, tanah, air, dan udara. Tanaman menjadi rusak, logam-logam berat akan diserap di tanah, serta merusak material bangunan karena hujan asam bersifat korosif. Serta kebocoran lapisan ozon sehingga membuat keseimbangan ekosistem jadi terganggu akibat efek rumah kaca. Polusi udara yang tidak terkendali menyebabkan panas yang seharusnya dipantulkan ke luar angkasa tetapi justru terperangkap oleh gas-gas karbon dioksida, metan, sulfur dioksida dan gas rumah kaca yang nantinya akan menjadi pemanasan global (*global warning*). Dan menyebabkan hujan asam karena oksida belerang dan oksida nitrogen hasil pembakaran batu bara yang ada keudara bereaksi dengan uap air membentuk awan asam (asam sulfat dan asam nitrat) [1].

Untuk menanggulangi masalah polusi di perkotaan dapat mengurangi penggunaan bahan bakar fosil pada industri, pembangkit listrik, dan rumah tangga, memanfaatkan energy alternative yang ramah lingkungan seperti biogas, energy surya atau energy panas bumi. Melakukan pengawasan lebih ketat di wilayah hutan yang rawan terbakar. Serta dapat melakukan dengan keberadaan jumlah pohon yang banyak untuk menanggulangi pencemaran udara yang kurang bersih. Pohon dapat mengurangi polusi udara, dan menanam pohon adalah salah satu tindakan yang paling tepat untuk dilakukan dalam membantu kualitas udara. Pohon akan menyerap karbon dioksida yang

menjadi alasan dari polusi udara, dan pohon akan menghasilkan oksigen yang baik untuk tubuh manusia.

Maka dari itu, penelitian ini akan menggunakan citra udara *Red-Green-Blue* (RGB) *channel* dengan resolusi tinggi sebagai citra masukan yang akan diubah menjadi format *Hue-Saturation-Value* (HSV) *channel* untuk mencari koordinat dengan mempertimbangkan spektrum warna. Metode *Oriented FAST and Rotated BRIEF* (ORB) untuk mencari *keypoint* titik-titik yang mengandung banyak informasi yang dihasilkan dari metode ekstraksi fitur *Feature from Accelerated Segment Test* (FAST). Serta memanfaatkan *BRIEF feature descriptor* yang telah dikembangkan sebagai output. Metode ORB untuk membedakan area hijau tanaman dan bukan area hijau tanaman yang terdapat di suatu daerah urban yang nantinya akan menjadi pertimbangan kebijakan perencanaan tata kota dan wilayah pada bidang lahan hijau bagi pemerintah.

1.2. Tujuan

Tujuan dari pembuatan penelitian ini yaitu membuat sistem yang dapat mendeteksi area tanaman hijau yang terdapat pada citra udara dengan resolusi sangat tinggi menggunakan metode *Oriented FAST and Rotated BRIEF*.

1.3. Manfaat

Manfaat yang diharapkan dari pengerjaan penelitian tugas akhir ini adalah:

1. Dapat mendeteksi keberadaan area hijau di daerah urban melalui citra udara di daerah perkotaan.
2. Bagi pemerintah kota sebagai dasar pertimbangan kebijakan perencanaan tata kota dan wilayah pada bidang lahan hijau.

1.4. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam penelitian ini antara lain:

1. Bagaimana membedakan area hijau dengan objek lain (bukan tanaman hijau) yang memiliki warna hijau yang sama?
2. Bagaimana mengimplementasikan deteksi area hijau menggunakan nilai *Hue Saturation Value* (HSV)?
3. Bagaimana mengimplementasikan metode Ekstraksi *Oriented FAST and Rotated BRIEF* (ORB) untuk mendeteksi area hijau?

1.5. Batasan Permasalahan

Permasalahan yang dibahas dalam penelitian ini memiliki beberapa batasan, antara lain:

1. Gambar masukan program yang digunakan adalah wilayah perkotaan Sapporo, Jepang pada tahun 2010.
2. Gambar masukan berekstensi png.
3. Output dari implementasi ini berupa penanda posisi area hijau.
4. Menggunakan nilai HSV dan metode Ekstraksi *Oriented FAST and Rotated BRIEF* (ORB).
5. Pembangunan perangkat lunak menggunakan bahasa pemrograman python.

1.6. Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Penyusunan Proposal

Pada awal Tugas Akhir ini adalah menyusun proposal Tugas Akhir. Pada proposal diajukan gagasan untuk mengimplementasikan deteksi area hijau pada daerah urban menggunakan ekstraksi fitur *Oriented FAST and Rotated BRIEF* (ORB).

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu mengenai bahasa pemrograman python, nilai HSV, operasi morfologi, dan ekstraksi fitur *Oriented FAST and Rotated BRIEF*.

3. Implementasi

Dalam penelitian ini nantinya akan menghasilkan gambar dengan tanda area hijau yang terdeteksi. Berikut beberapa hal yang diperlukan dalam implementasi:

- a. Python 2.7.11
- b. Pustaka OpenCV 3.0
- c. Pustaka Numpy 1.9.2
- d. Pustaka Matplotlib 1.3.0

4. Pengujian dan evaluasi

Pengujian dari aplikasi ini akan dilakukan dengan menggunakan pengujian blackbox. Pengujian blackbox adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. Pengujian ini dilakukan untuk menguji apakah posting terkelompokkan ke dalam kategori yang sesuai atau tidak.

5. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan proses dokumentasi dan pembuatan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan penelitian.

1.7. Sistematika Penulisan

Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini.

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan penelitian, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan penelitian.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan penelitian ini.

Bab III Analisis dan Perancangan Sistem

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka aplikasi.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan dan implementasi fitur-fitur penunjang aplikasi.

Bab V Pengujian dan Evaluasi

Bab ini membahas pengujian dengan metode kotak hitam (*black box testing*) untuk mengetahui aspek nilai fungsionalitas dari perangkat lunak dan nilai kegunaan yang dibuat dengan juga memperhatikan ketertarikan pada calon partisipan untuk menggunakan aplikasi ini.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan penelitian ini.

Lampiran

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Penelitian. Teori-teori tersebut meliputi pengertian dan beberapa analisis terkait pendeteksian area tanaman hijau di daerah urban.

2.1. Citra Udara

Definisi citra (*image*) adalah suatu representasi, kemiripan atau imitasi dari suatu objek atau benda. Secara harfiah, citra adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi. Citra yang terlihat merupakan cahaya yang direfleksikan dari objek, sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap oleh alat-alat optik, misalnya mata manusia, kamera, sensor satelit dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.



Gambar 2.1 Citra udara Kota Sapporo, Jepang

Foto udara seperti pada Gambar 2.1 merupakan hasil pemotretan pada satu daerah yang memiliki ketinggian tertentu menggunakan sebuah kamera pada ruang lingkup atmosfer. Contohnya, seperti proses pemotretan yang dilakukan di atas pesawat terbang, balon udara, helikopter, *drone*, dan berbagai wahana yang mengudara lainnya.

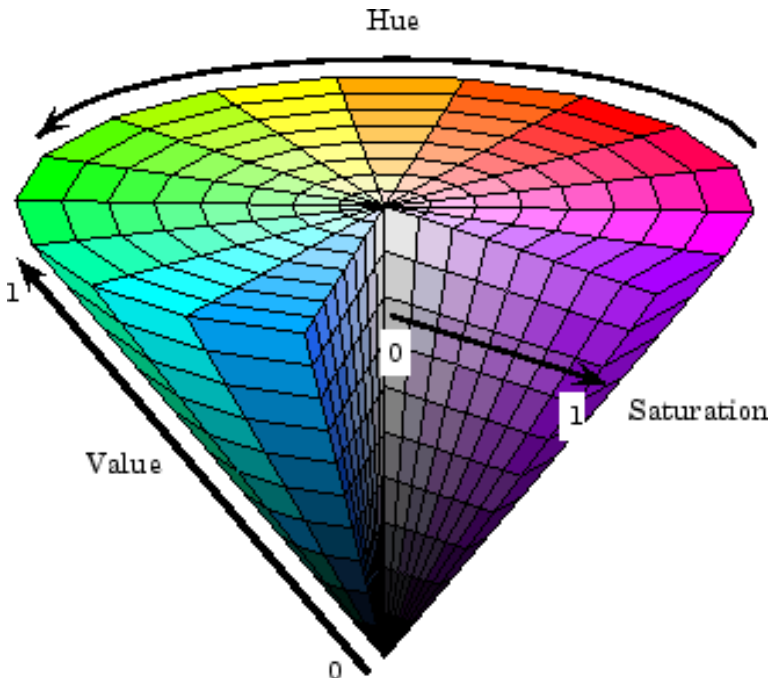
Keuntungan menggunakan foto udara yaitu memberikan hasil gambar atau menciptakan citra yang jauh lebih baik dan detail (resolusi sekitar 15 cm), tidak terkena awan, karena sistem pengoperasian berada di bawah awan. Sedangkan untuk kelemahan dari foto udara adalah terbangun atas berbagai kumpulan scene kecil yang sangat banyak, sistem pemotretan pada area yang jauh lebih luas dengan yang lain. Sistem pengoperasian foto udara tergantung dengan faktor cuaca dan angin. Contohnya, ketika menggunakan UAV memberikan hasil foto udara yang kurang stabil dan bagus jika dihempas tiupan angin yang kencang, hasilnya pun kurang optimal dan stabil.

Kelemahaan yang lain menggunakan foto udara, foto udara harus disertai dengan pengambilan *ground point control* lapangan untuk melaksanakan proses koreksi geometrik. Jika tidak dilakukan seperti ini, hasilnya kurang akurat dan sangat rendah dari sisi geometrik. Jika dilihat dari biaya yang dikeluarkan foto udara lebih mahal dibandingkan menggunakan citra satelit karena banyak kebutuhan yang diperlukan. Mulai dari izin penerbangan jika menggunakan helikopter atau pesawat terbang, biaya operasional pesawat, biaya lapangan, pengambilan koordinat GCP dan masih banyak yang lainnya.

2.2. HSV Image

HSV mendefinisikan warna dalam terminologi *Hue* *Saturation* dan *Value*. Keuntungan HSV adalah terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia. *Hue* menyatakan informasi warn-warna dasar, seperti merah, violet, dan kuning. HSV digunakan untuk menentukan kemerahan

(*redness*), kehijauan (*greenness*). *Saturation* adalah kemurnian atau kekuatan warna. Sedangkan *Value* adalah kecerahan dari warna. Apabila nilai *Value* 0 maka warna yang ditampilkan akan menjadi hitam, semakin besar nilai *Value* maka semakin cerah dan muncul variasi-variasi baru dari warna yang ditampilkan [3].

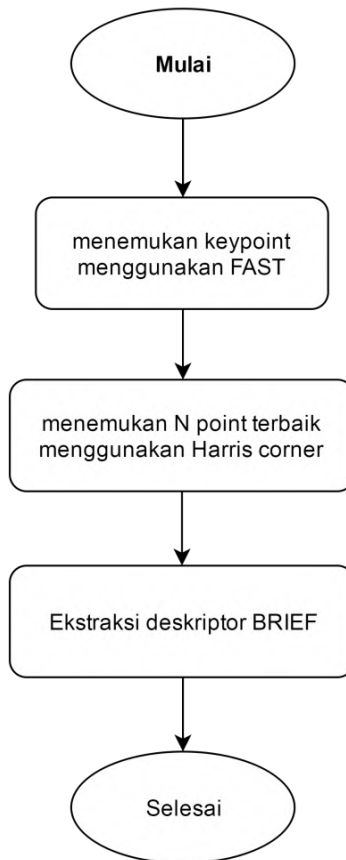


Gambar 2.2 Contoh model HSV

2.3. Oriented FAST and Rotated BRIEF

Oriented FAST and Rotated BRIEF (ORB) merupakan algoritma yang umum digunakan untuk ekstraksi fitur. ORB menggunakan *Feature from Accelerated Segment Test* (FAST) untuk melakukan deteksi titik fitur dan memanfaatkan BRIEF *feature descriptor* yang telah dikembangkan sebagai output.

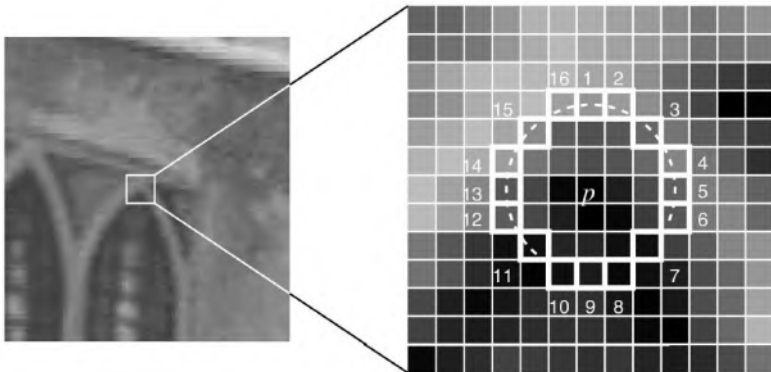
Proses yang dilakukan oleh ORB dapat dilihat pada Gambar 2.3. Kelebihan dari algoritma ini adalah kecepatannya dalam melakukan ekstraksi fitur. Fitur yang dihasilkan dari metode ini berupa *binary descriptor*, sehingga mempermudah pencocokan pada proses selanjutnya. Selain itu, ORB juga memiliki invarian rotasi dan mengurangi kepekaan terhadap noise [4].



Gambar 2.3 Diagram alir ORB

2.3.1. FAST Orientation

Feature from Accelerated Segment Test (FAST) digunakan karena sifat komputasi yang dimiliki FAST. FAST mendeteksi *keypoint* pada gambar, FAST mengambil satu parameter, intensitas *threshold* antara pusat pixel dengan lingkaran pixel disekitar pusat pixel, ilustrasi dapat dilihat pada Gambar 2.4.



Gambar 2.4 Deteksi keypoint menggunakan FAST

Kriteria uji segmen beroperasi dengan memper-timbangkan lingkaran 16 pixel disudut calon p . Algoritma ini dijelaskan sebagai berikut:

1. Pilih pixel di p pada gambar. Asumsikan intensitas pixel ini menjadi I_p . Ini adalah pixel yang diidentifikasi sebagai titik tujuan atau tidak.
2. Set batas nilai intensitas T .
3. Pertimbangkan lingkaran 16 pixel sekitar pixel p . (lingkaran Bresenham dari radius 3).
4. “ N ” pixel yang berdekatan dari 16 dibutuhkan untuk menjadi nilai diatas atau dibawah nilai dari I_p .
5. Untuk membuat algoritma cepat, pertama mem-bandingkan intensitas pixel 1, 5, 9, dan 13 dari lingkaran dengan I_p . Seperti terlihat dari gambar diatas, setidaknya

tiga dari empat pixel ini harus memenuhi kriteria nilai threshold sehingga titik menarik (*interest point*) menjadi ada.

6. Jika setidaknya tiga dari empat nilai-nilai pixel $-I_1, I_5, I_9, I_{13}$ adalah tidak diatas atau dibawah $I_p + T$, Maka P bukanlah titik tujuan (sudut). Dalam hal ini menolak p pixel sebagai titik tujuan. Lain jika setidaknya tiga dari pixel berada diatas atau dibawah $I_p + T$, kemudian memeriksa semua 16 pixel dan periksa apakah 12 pixel berdekatan dalam kriteria.
7. Ulangi prosedur untuk semua pixel dalam gambar [5].

Detektor ini sendiri menunjukkan kinerja yang tinggi, namun ada beberapa kelemahan, yaitu:

- Kecepatan tinggi tidak dapat ditolah sebanyak kandidat $n < 12$, Karena titik dapat menjadi sudut ketika hanya dua yang keluar dari 4 pixel yang secara bersama menjadi signifikan terang atau secara bersama menjadi signifikan gelap dari p (dengan asumsi pixel berdekatan).
- Efisiensi detector akan tergantung pada urutan dan distribusi sudut penampilan. Hal ini tidak mungkin bahwa ini adalah pilihan dari pixel yang optimal
- Beberapa fitur yang terdeteksi berdekatan satu sama lain [6].

FAST tidak menghasilkan ukuran sudut, tetapi FAST memiliki respon besar terhadap tepi. Harris corner digunakan untuk memerintakan *keypoint* pada FAST. Harris corner menetapkan threshold yang cukup rendah untuk mendapatkan lebih banyak N *keypoint*. Kemudian mengurutkan berdasarkan berdasarkan ukuran Harris. Setelah itu memilih N *keypoint* terbaik [4].

2.3.2. Rotated BRIEF

BRIEF descriptor adalah deskripsi bit string dari gambar yang dibangun dari intensitas biner. Sudut *keypoint* berorientasi menyajikan penampilan yang lebih seragam pada biner. Varians yang tinggi membuat fitur lebih diskriminatif, karena merespon secara berbeda untuk input. BRIEF sangat tergantung pada orientasi acak pada *keypoints* [4].

2.4. Morfologi Citra

Operasi Morfologi adalah teknik pengolahan citra yang didasarkan pada bentuk segmen atau *region* dalam citra. Karena difokuskan pada bentuk objek, maka operasi ini diterapkan pada citra biner. Biasanya segmen tadi didasarkan pada objek yang menjadi perhatian.

Segmentasi dilakukan dengan membedakan antara objek dan latar, antara lain dengan memanfaatkan operasi pengambangan yang mengubah citra warna dan skala keabuan menjadi citra biner. Nilai biner dari citra hasil merepresentasikan dua keadaan, yaitu objek dan bukan objek (latar).

Hasil operasi morfologi dapat dimanfaatkan untuk pengambilan keputusan dengan analisis lebih lanjut. Operasi ini antara lain meliputi dilasi, erosi, penutupan (*closing*), dan pembukaan (*opening*).

2.4.1. Structuring Element

Ide dasar dari operasi morfologi biner adalah untuk memeriksa gambar secara sederhana, bentuk yang telah ditentukan, menarik kesimpulan tentang bagaimana bentuk ini cocok atau tidak jika dibandingkan dengan bentuk dalam gambar. Bentuk pada pemeriksaan sederhana ini disebut "*structuring element*", yang merupakan suatu citra biner. *Structuring Element* dapat diibaratkan dengan *mask* pada pemrosesan citra yang lain selain pemrosesan citra secara morfologi.

2.4.2. Dilasi

Dilasi adalah operasi morfologi yang akan menambahkan pixel pada batas antar objek dalam suatu citra digital. Dilasi merupakan suatu proses menambahkan piksel pada batasan dari objek dalam suatu citra sehingga nantinya apabila dilakukan operasi ini maka citra hasilnya lebih besar ukurannya dibandingkan dengan citra aslinya. Dalam operasi dilasi, citra asli akan mengalami pelebaran dengan mengikuti bentuk *Structuring Element* yang digunakan. Secara matematis, dilasi A oleh B didefinisikan seperti pada persamaan (2.1)

$$A \oplus B = \cup_{b \in B} A_b \quad (2.1)$$

Gambar 2.5 merupakan hasil citra setelah dilakukan proses Dilasi dari citra asli. Warna hijau merupakan piksel tambahan setelah dilakukan Dilasi.

2.4.3. Erosi

Erosi merupakan kebalikkan dari dilasi. Proses ini akan membuat ukuran sebuah citra menjadi lebih kecil. Berbeda dengan dilatasi, apabila erosi dilakukan maka yang dikerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari piksel yang ditambah atau dihilangkan bergantung pada ukuran dan bentuk dari *structuring element* yang digunakan untuk memproses image tersebut. Erosi A oleh B didefinisikan oleh persamaan (2):

$$A \ominus B = \{z \in E \mid B_z \subseteq A\} \quad (2.2)$$

Gambar 2.6 merupakan hasil citra setelah dilakukan proses erosi dari citra asli. Warna hijau merupakan piksel yang dihilangkan setelah dilakukan proses erosi, sehingga piksel hasilnya hanya yang berwarna hitam.

2.4.4. *Opening*

Opening merupakan kombinasi proses dimana suatu citra digital dikenai operasi erosi dilanjutkan dengan dilasi. Operasi opening pada citra mempunyai efek memperhalus batas-batas objek, memisahkan objek-objek yang sebelumnya bergandengan, dan menghilangkan objek-objek yang lebih kecil daripada ukuran *structuring element*. Operasi *opening* dari himpunan A oleh *Structuring Element* B , yang dinotasikan dengan $A \circ B$, didefinisikan pada persamaan 3

$$A \circ B = \{ A \ominus B \} \oplus B \quad (2.3)$$

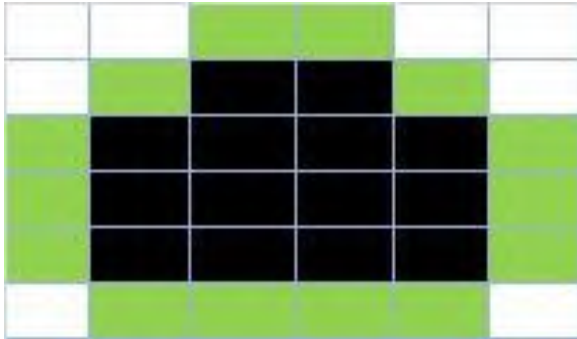
Gambar 2.7 merupakan hasil citra setelah dilakukan proses Opening dari citra asli. Proses yang dilakukan untuk mendapatkan opening adalah dengan cara melakukan operasi erosi, lalu hasil erosi tersebut di dilasi lagi.

2.4.5. *Closing*

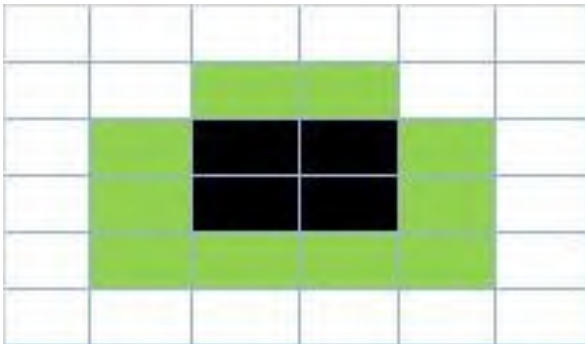
Closing merupakan kombinasi dimana suatu citra dikenai operasi dilasi dilanjutkan dengan erosi. Operasi closing juga cenderung akan memperhalus objek pada citra, namun dengan cara menyambung pecahan-pecahan dan menghilangkan lubang-lubang kecil pada objek. Operasi *closing* dari himpunan A oleh *Structuring Element* B yang dinotasikan dengan $A \cdot B$, didefinisikan pada persamaan 4

$$A \cdot B = \{ A \oplus B \} \ominus B \quad (2.4)$$

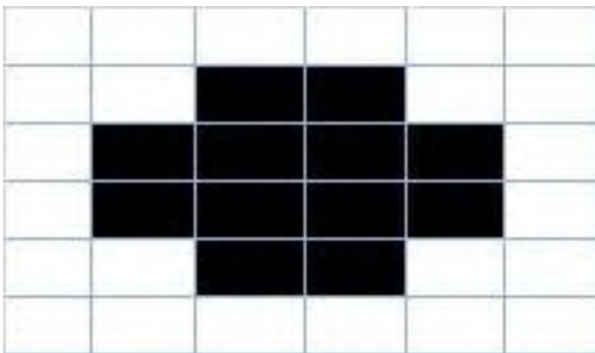
Gambar 2.8 merupakan hasil citra setelah dilakukan proses Closing dari citra asli. Proses yang dilakukan untuk mendapatkan Closing adalah dengan cara melakukan operasi dilasi, lalu hasil dilasi tersebut dilakukan proses erosi lagi [7].



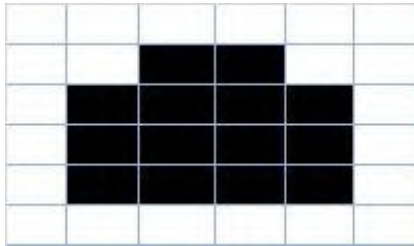
Gambar 2.5. Contoh Operasi Dilasi



Gambar 2.6. Contoh Operasi Erosi



Gambar 2.7. Contoh Operasi Opening



Gambar 2.8. Contoh Operasi Closing

2.5. Perhitungan Kinerja Aplikasi

Perhitungan kinerja aplikasi pada sistem ini adalah dengan menggunakan akurasi. Akurasi yang digunakan untuk mengukur tingkat kualitas deteksi area tanaman hijau. Kategori uji coba dinyatakan benar jika hasil deteksi adalah tanaman hijau. Sedangkan kategori uji coba dinyatakan salah jika hasil deteksi bukanlah objek tanaman hijau. Persamaan untuk pengukuran akurasi dapat dilihat pada Persamaan (2.5) dan Persamaan (2.6).

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \quad (2.5)$$

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \quad (2.6)$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.7)$$

Pada persamaan diatas, *True positive* pada pengujian ini adalah area tanaman hijau yang terdeteksi pada citra. *False positive* menyatakan penanda yang berada pada objek selain tanaman hijau. *False negative* menyatakan tanaman hijau yang tidak terdeteksi oleh aplikasi. *F1* digunakan untuk mendapatkan satu nilai dari *precision* atau *recall* yang mencerminkan kualitas. [8]

[Halaman ini sengaja dikosongkan]

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada Bab 3 ini akan dijelaskan mengenai analisis dan perancangan perangkat lunak untuk mencapai tujuan dari penelitian ini. Perancangan ini meliputi perancangan data, perancangan proses, dan perancangan antar muka, serta juga akan dijelaskan tentang analisis implementasi metode secara umum pada sistem.

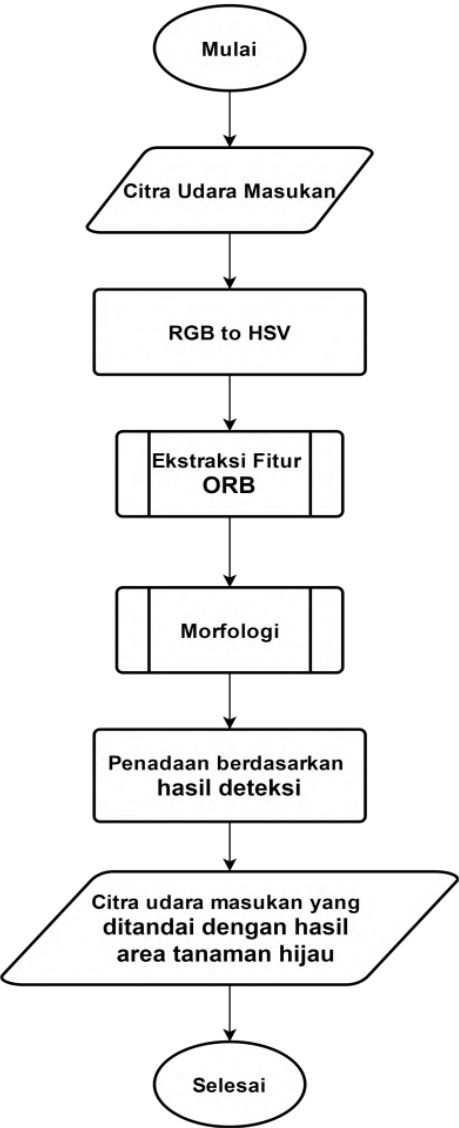
3.1. Desain Umum Sistem

Pada penelitian ini akan dibangun sebuah sistem untuk melakukan pendeteksi area hijau di wilayah udara perkotaan dengan data masukan citra udara yang beresolusi sangat tinggi dengan menggunakan pustaka OpenCV. Area hijau yang dideteksi adalah tanaman hijau yang berwarna hijau, bukan atap gedung dan bukan tanaman berwarna kuning atau berwarna coklat atau warna kemerah-merahan.

Tahap pertama dimulai dengan konversi nilai *Red-Green-Blue* (RGB) channel diubah menjadi *Hue-Saturation-Value* (HSV) channel. Tahap berikutnya adalah *thresholding* nilai HSV, tujuannya untuk mendapatkan hanya warna hijau pada citra masukan.

Selanjutnya dilakukan ekstraksi fitur, metode yang digunakan adalah Oriented FAST and Rotated BRIEF (ORB). Pada tahap ini, citra masukan yang telah mengalami *thresholding* kemudian dideteksi keberadaan *interest point* atau titik-titik yang mengandung banyak informasi.

Tahap ketiga adalah proses morfologi closing dan dilasi, tujuan dari tahap ini adalah untuk mempermudah ketika area tanaman hijau ditandai oleh sistem. Tahap terakhir yaitu melakukan penandaan terhadap *interest point* yang sudah dilakukan morfologi. Alur metode secara umum akan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram alur Proses program

3.2. Data

Pada subbab ini akan dijelaskan mengenai data yang akan digunakan sebagai masukan perangkat lunak untuk selanjutnya diolah dan dilakukan pengujian sehingga menghasilkan data keluaran yang diharapkan.

3.2.1. Data Masukan

Data masukan adalah data yang akan diolah oleh sistem untuk mendapatkan hasil keluaran yang sudah ditentukan sistem. Pada sistem ini, data yang digunakan berupa citra udara perkotaan Sapporo, Jepang pada tahun 2010 yang merupakan area yang mengandung tanaman hijau, banyak bangunan betingkat pencakar langit dan juga mengandung jalanan yang landai. Pada sistem ini akan menggunakan 50 citra udara, masing-masing citra memiliki resolusi 1.000 pixel x 1.000 pixel dengan pita warna *Red-Green-Blue* (RGB). Contoh citra masukan yang digunakan sebagai data masukan ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Contoh citra udara masukan

3.2.2. Data Proses

Data proses adalah data yang digunakan selama proses berjalannya sistem yang merupakan hasil pengolahan dari data masukan untuk diproses kembali menjadi data keluaran di tahap selanjutnya. Data proses yang diguna-kan dalam sistem ini ditunjukkan oleh Tabel 3.1 sebagai berikut.

Tabel 3.1 Data Proses

No.	Nama Data	Keterangan
1	img	Matriks citra udara masukan yang memiliki pita warna BGR
2	img_RGB	Matriks citra udara yang sudah diubah ke citra RGB
3	img_hsv	Matriks citra udara yang sudah diubah ke citra HSV
4	img_gray	Matriks citra udara yang sudah diubah ke citra <i>grayscale</i>
5	lower_green	Batas bawah nilai HSV hijau
6	upper_green	Batas atas nilai HSV hijau
7	mask	Thresholding nilai HSV hijau
8	orb	Objek pendeteksi dan ekstraksi fitur ORB
9	kp	Keypoint yang terdeteksi ORB
10	des	Fitur descriptor yang dideteksi ORB melalui kp
11	Kernel	Structure element yang digunakan untuk morfologi
12	mask	Matriks citra biner hasil morfologi <i>closing</i>
13	Erosion	Matriks citra biner hasil morfologi erosi
14	Dilation	Matriks citra biner hasil morfologi <i>dilation</i>
15	ctr	<i>Keypoint</i> yang berada didalam <i>contour</i>

3.2.3. Data Keluaran

Data keluaran pada sistem ini berupa citra udara masukan dengan tambahan deteksi area tanaman hijau. Area hijau yang dideteksi adalah tanaman hijau yang berwarna hijau, bukan atap gedung dan bukan tanaman berwarna kuning atau berwarna coklat atau warna kemerah-merahan. Contoh data keluaran dapat dilihat pada Gambar 3.3.

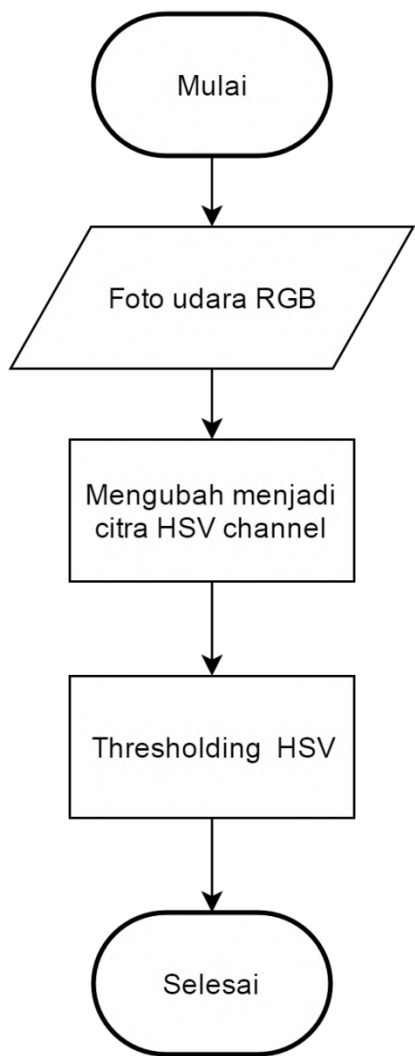


Gambar 3.3 Contoh citra udara hasil deteksi

3.3. Perancangan Proses

Pada subbab ini akan dibahas mengenai perancangan proses yang dilakukan untuk memberikan gambaran secara rinci pada setiap alur implemtasi metode pada sistem deteksi area tanaman hijau. Alur tersebut nantinya akan digunakan dalam tahap implementasi program.

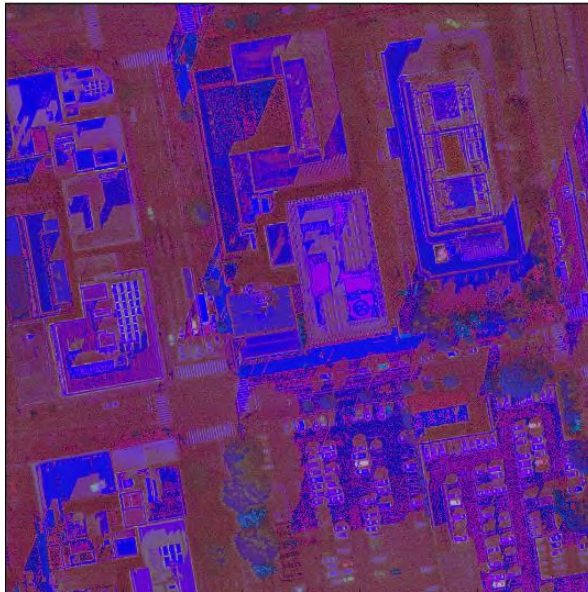
3.3.1. Konversi HSV



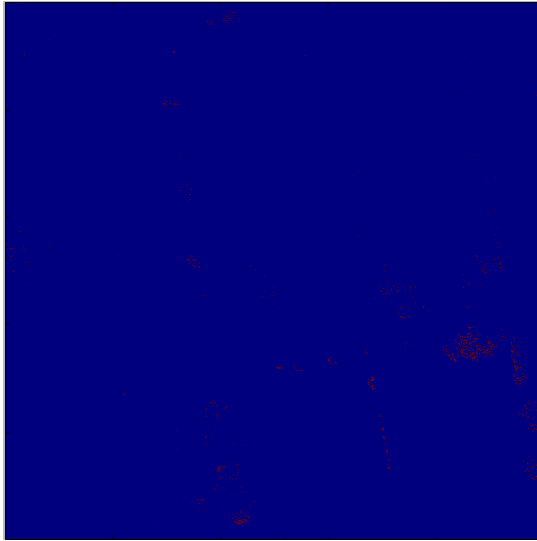
Gambar 3.4 Alur proses konversi HSV

Tahap awal dimulai dengan konversi nilai *Red-Green-Blue* (RGB) channel diubah menjadi *Hue-Saturation-Value* (HSV) channel seperti pada Gambar 3.5 Contoh citra udara konversi HSV. Tahap berikutnya adalah *thresholding* nilai HSV, tujuannya untuk mendapatkan hanya warna hijau pada citra masukan, Pada sistem ini menggunakan batas minimal warna hijau dengan nilai HSV (50,50,50) sampai dengan maksimal warna hijau dengan nilai HSV (60,255,255). Alur pemrosesan tahap ini dapat dilihat pada Gambar 3.4. Hasil *thresholding* nilai HSV dapat dilihat pada Gambar 3.6.

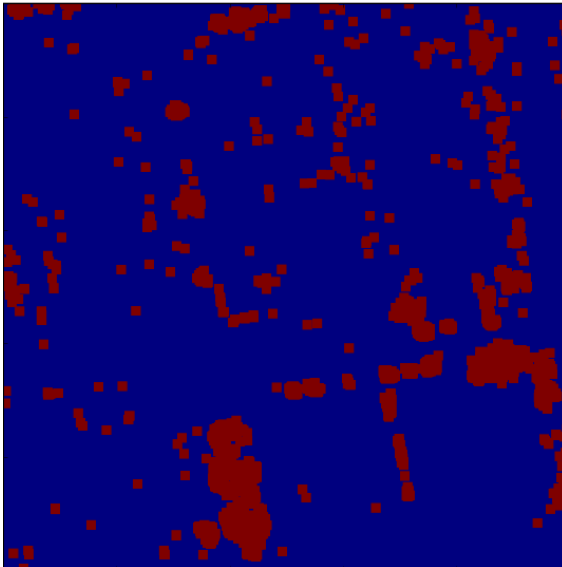
Jika diperjelas menggunakan morfologi akan terlihat seperti Gambar 3.7. Namun sistem ini tidak melakukan tahap morfologi ini setelah *thresholding* nilai HSV, tujuan morfologi dilakukan agar memperjelas output dari hasil *thresholding* HSV channel.



Gambar 3.5 Contoh citra udara konversi HSV



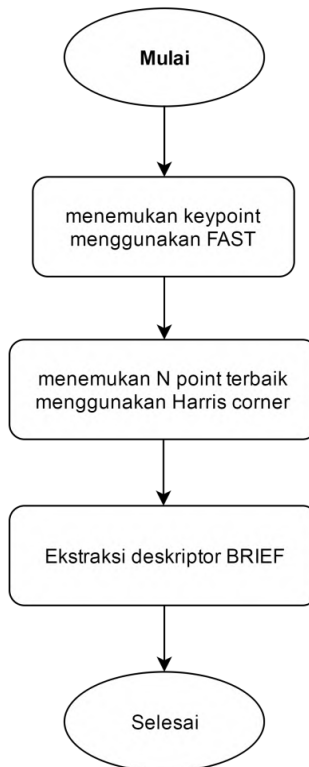
Gambar 3.6 Contoh citra udara hasil thresholding HSV



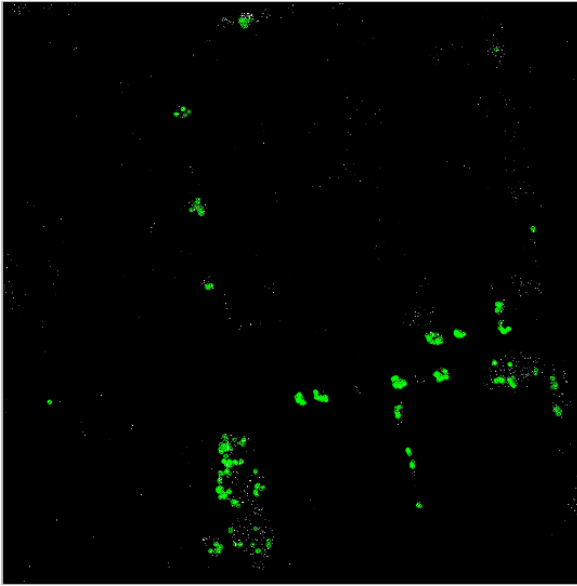
Gambar 3.7 Contoh citra hasil thresholding HSV yang diperjelas

3.3.2. Ekstraksi Fitur ORB

Kemudian dilanjutkan dengan pencarian *Keypoint*, metode yang digunakan adalah *Oriented FAST and Rotated BRIEF*. *Keypoint* atau titik-titik yang mengandung banyak informasi yang dihasilkan dari metode ini digunakan dalam tahap ekstraksi fitur menggunakan *Feature from Accelerated Segment Test* (FAST). Serta memanfaatkan BRIEF *feature descriptor* yang telah dikembangkan sebagai output, hasil dari output berupa *binary descriptor*. Alur proses pada tahap ini dapat dilihat pada Gambar 3.8. Hasil ekstraksi fitur ORB dapat dilihat pada Gambar 3.9.



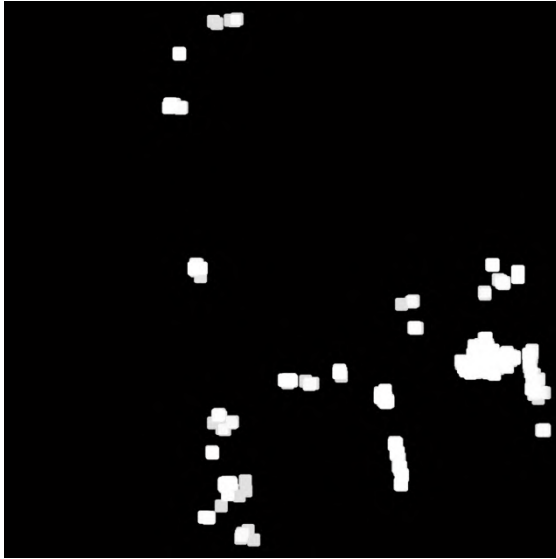
Gambar 3.8 Alur proses ORB



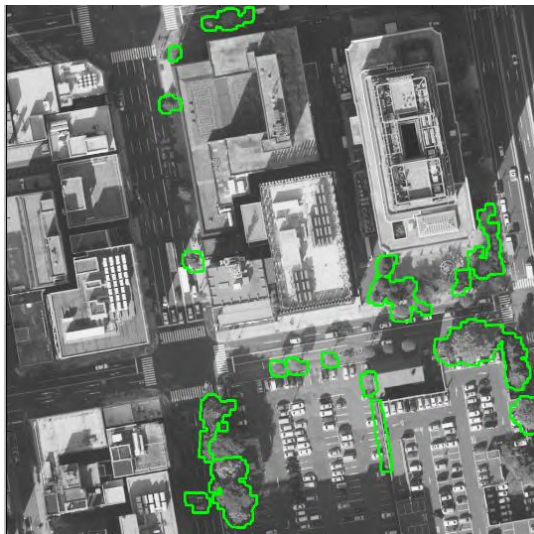
Gambar 3.9 Contoh citra udara hasil ekstraksi fitur ORB

3.3.3. Morfologi

Setelah dilakukan ekstraksi fitur, dilakukan *closing morphologi* untuk memperhalus objek pada citra dengan cara menyambungkan pecahan-pecahan dan menghilangkan lubang-lubang kecil pada citra objek. Selain *closing morphologi* dilakukan *dilation morphologi*, tujuan dari tahap ini supaya menghilangkan titik-titik *noise* pada citra sehingga didapat citra yang lebih bersih. Hasil dari *closing morphologi* dan *dilation morphologi* dilihat pada Gambar 3.10. Kemudian dilanjutkan dengan penandaan area hijau dari hasil deteksi dengan latar citra asli dengan pita warna *grayscale*. Penandaan area ini dilakukan dengan mendeteksi tepi *keypoint*. Contoh data keluaran tersebut dapat dilihat pada Gambar 3.11.



Gambar 3.10 Contoh citra udara hasil morfologi



Gambar 3.11 Contoh citra udara setelah ditandai area hijau

3.4. Perancangan Antarmuka Perangkat Lunak

Pada bagian ini membahas mengenai perancangan antarmuka pada sistem. Secara umum terdapat 1 halaman antarmuka pada sistem ini, yaitu halaman citra keluaran yang sudah terdeteksi dengan area tanaman hijau. Pada halaman ini akan ditampilkan seperti pada Gambar 3.12.

Citra pada kotak berjudul “Gambar Original” adalah citra udara masukan. Citra pada kotak berjudul “Hasil” adalah g citra udara keluaran yang sudah terdeteksi area tanaman hijau.

Pada bagian bawah halaman *Output*, terdapat beberapa tombol yang dapat memudahkan pengguna untuk melihat citra pada setiap kotak. Nama dan fungsi setiap tombol dapat dilihat pada Tabel 3.2.

Tabel 3.2. Nama dan Fungsi Tombol

No.	Nama Tombol	Simbol	Fungsi
1	<i>Reset Original View</i>		Mengembalikan halaman utama menjadi tampilan awal
2	<i>Back to Previous View</i>		Mengembalikan tampilan ke satu tampilan sebelum
3	<i>Forward to Next View</i>		Mengubah tampilan ke tampilan setelah
4	<i>Pan / Zoom</i>		Mengaktifkan mode pan and zoom (klik kiri <i>mouse</i> lalu tahan untuk <i>zooming</i> , klik kanan <i>mouse</i> lalu tahan untuk menggeser gambar)
5	<i>Zoom to Rectangle</i>		Memperbesar gambar sesuai area persegi panjang yang digambar
6	<i>Configure Subplot</i>		Menampilkan menu untuk menyesuaikan jarak setiap <i>subplot</i>
7	<i>Save</i>		Menyimpan gambar yang ditampilkan pada halaman utama



Gambar 3.12 Rancangan Halaman Antarmuka Perangkat Lunak

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

Bab ini membahas implementasi dari perancangan sistem sesuai dengan perancangan yang telah dibuat. Bahasa pemrograman yang digunakan untuk implementasi sistem adalah bahasa pemrograman Python dengan pustaka OpenCV 3.0, Numpy dan Matplotlib.

4.1. Lingkungan Implementasi

Lingkungan implementasi sistem yang digunakan untuk mengembangkan tugas akhir memiliki spesifikasi perangkat keras dan perangkat lunak seperti ditampilkan pada Tabel 4.1

Tabel 4.1 Lingkungan Implementasi Sistem

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel® Core™ i3-4150 CPU @ 3.50GHz
	Memori	4 GB
Perangkat Lunak	Sistem Operasi	Windows 8 64-bit
	Perangkat Pengembang	IDLE (Python GUI)

4.2. Implementasi Proses

Implementasi proses dilakukan berdasarkan perancangan proses yang sudah dijelaskan pada bab analisis dan perancangan.

4.2.1. Implementasi Tahap Thresholding HSV

Pada subbab ini akan dilakukan beberapa tahap operasi thresholding, yaitu tahap mengubah citra RGB *channel* ke HSV *channel*, kemudian penerapan *dilation transform morphology*. Penerapan algoritma implemena *hsv* dapat dilihat pada Kode Sumber 4.1 Kode program untuk thresholding Gambar HSV

1	<code>import cv2</code>
2	<code>import numpy as np</code>
3	<code>import matplotlib.pyplot as plt</code>
4	
5	<code>img = cv2.imread</code> <code>('D: Sapporo/split/01_006.png')</code>
6	<code>img_RGB = cv2.cvtColor(img,</code> <code>cv2.COLOR_BGR2RGB)</code>
7	<code>img_hsv = cv2.cvtColor(img,</code> <code>cv2.COLOR_BGR2HSV)</code>
8	<code>img_gray = cv2.cvtColor(img,</code> <code>cv2.COLOR_BGR2GRAY)</code>
9	<code>img_RGB2 = cv2.cvtColor(img_gray,</code> <code>cv2.COLOR_GRAY2RGB)</code>
10	
11	<code># define range of green color in HSV</code>
12	<code>lower_green = np.array([50,50,50])</code>
	<code>upper_green = np.array([60,255,255])</code>
13	
14	<code># Threshold the HSV image to get only green</code> <code>color</code>
15	<code>mask = cv2.inRange(img_hsv, lower_green,</code> <code>upper_green)</code>

Kode Sumber 4.1 Kode program untuk thresholding Gambar HSV

Fungsi `cv2.imread()`, Fungsi `cv2.cvtColor()`, dan Fungsi `cv2.inRange()` merupakan bawaan dari pustaka OpenCV. Fungsi `cv2.imread()` digunakan untuk memuat citra. Kembalian dari fungsi ini berupa matriks gambar dengan pita warna BGR (*Blue-Green-Red*). Fungsi `cv2.cvtColor()` digunakan untuk mengubah pita warna citra menjadi pita warna yang diinginkan. Fungsi `cv2.inRange()` digunakan untuk thresholding citra HSV dengan warna yang diinginkan. Pada sistem ini menggunakan batas minimal warna hijau dengan nilai HSV (50,50,50) sampai dengan maksimal warna hijau dengan nilai HSV (60,255,255).

4.2.2. Implementasi Tahap ekstraksi fitur ORB

Setelah mendapatkan nilai HSV *Channel*, pada subbab ini membahas implementasi tahap pencarian *keypoint*. Implementasi tahap pencarian *keypoint* dapat dilihat pada Kode Sumber 4.2 Kode Program mencari Keypoint dengan ORB

1	# bitwise-And mask and original image
2	res = cv2.bitwise_and(img,img, mask=mask)
3	# FAST feature and Harris Corner
4	orb = cv2.ORB_create(edgeThreshold=20, patchSize=20, nlevels=8, fastThreshold=5, scaleFactor=1.2, WTA_K=2, scoreType= cv2.ORB_HARRIS_SCORE, firstLevel=0, nfeatures=500)
5	# find the keypoints with ORB
6	kp = orb.detect(mask, None)
7	# compute the descriptors with ORB
8	kp, des = orb.compute(mask, kp)

Kode Sumber 4.2 Kode Program mencari Keypoint dengan ORB

Fungsi *cv2.ORB_create()*, Fungsi *orb.detect()*, dan Fungsi *orb.compute()* merupakan bawaan dari pustaka OpenCV. Pembuatan objek detector ORB menggunakan fungsi *cv2.ORB_create()*. Didalam fungsi *cv2.ORB_create()* terdapat *edgeThreshold*, *patchsize*, *nlevels*, *fastThreshold*, *scaleFactor*, *WTA_K*, *scoreType*, *firstLevel*, *nfeatures*. Parameter pada fungsi ini bernilai masing-masing secara berurutan 25, 20, 9, 5, 1.2 ,2, *cv2.ORB_HARRIS_SCORE*, 0, dan 500.

Fungsi *orb.detect()* digunakan untuk mencari keypoint pada citra. Parameter dari fungsi ini adalah matriks citra masukan. Fungsi *orb.compute()* untuk ekstraksi fitur dari objek detector ORB yang telah dicari keypoint dari fungsi *orb.detect()*. Fungsi *compute* memiliki dua parameter, yaitu matriks citra dan *keypoints*.

4.2.3. Implementasi Tahap Morfologi

9	# Morphology
10	kernel = np.ones((5,5),np.uint8)
11	mask = cv2.morphologyEx(mask,cv2.MORPH_CLOSE, kernel)
12	erosion = cv2.erode(mask,kernel,iterations = 5)
13	dilation = cv2.dilate(mask,kernel,iterations = 4)

Kode Sumber 4.3 Kode program untuk morfologi gambar

4.2.4. Implementasi Tahap Pencocokan Fitur

Pada bagian ini membahas implementasi tahap pencocokan fitur. Implementasi tahap pencocokan fitur dapat dilihat pada Kode Sumber 4.4

1	ret,thresh = cv2.threshold(dilation,127,255,0)
2	im2, contours, hierarch = cv2.findContours (thresh,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)
3	flags = np.zeros(len(contours))
4	# pengecek Keypoint didalam contour
5	for i, contour in enumerate(contours):
6	for keypoint in kp:
7	ctr = cv2.pointPolygonTest(contour, keypoint.pt,False)
8	if ctr != -1 :
9	flags[i] = 1
10	# mencari gambar contour yg sudah ditandai
11	for i, contour in enumerate(contours):
12	if flags[i] == 1:
13	cv2.drawContours(img_RGB2, contour, -1, (0,255,0), 3)

Kode Sumber 4.4 Kode program untuk mencocokkan fitur

4.3. Impelementasi Antarmuka Perangkat Lunak

Implementasi tampilan antarmuka pada perangkat lunak ini berisi menampilkan citra asli dengan penanda tanaman hijau. Rancangan ini dapat dilihat pada Gambar 3.12 pada subbab 3.4. Implementasi tahap antarmuka perangkat lunak dapat dilihat pada Kode Sumber 4.5 Kode Program untuk menampilkan halaman Output

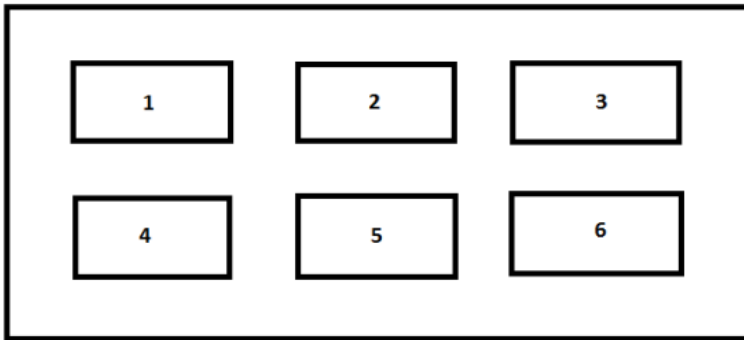
2	<code>plt.subplot(1,2,1), plt.imshow(img), plt.title("Gambar Original")</code>
3	<code>plt.subplot(1,2,2), plt.imshow(img_RGB2), plt.title("hasil")</code>
4	<code>mng = plt.get_current_fig_manager()</code>
5	<code>mng.window.state('zoomed')</code>
6	<code>plt.show()</code>

Kode Sumber 4.5 Kode Program untuk menampilkan halaman Output

Fungsi *subplot* digunakan untuk membagi tampilan pada halaman output menjadi beberapa kotak. Pada halaman ini, tampilan dibagi dua secara horizontal. Oleh karena itu, digunakan *subplot* 1x2. Parameter dari fungsi *subplot* terdiri dari tiga angka integer. Angka pertama menampilkan ukuran vertikal *subplot*. Angka berikutnya menampilkan ukuran Horizontal pada *subplot*. Sedangkan angka ketiga mempresentasikan indeks atau urutan *subplot* yang akan diakses ketika fungsi dijalankan. Sistem indeks pada *subplot* dapat dilihat pada Gambar 4.1.

Fungsi *get_current_fig_manager* digunakan untuk mengambil objek *figure manager* yang digunakan untuk menampilkan jendela saat ini. Objek *figure manager* ini diatur menjadi state *zoomed* agar jendela yang ditampilkan berukuran sebesar layar atau *maximized*.

Fungsi *show* digunakan untuk menampilkan jendela keluaran. Hasil dari implementasi halaman *output*, dapat dilihat pada Gambar 4.2.



Gambar 4.1 Visual Indeks pada subplot








Gambar 4.2 Hasil Implementasi Halaman Output

Gambar pada kotak disebelah kiri berjudul “Gambar Original” adalah gambar citra udara masukan dengan pita warna RGB. Gambar pada kotak disebelah kanan berjudul “Hasil” adalah gambar citra udara keluaran yang sudah terdeteksi area tanaman hijau.

Pada bagian bawah halaman *Output*, terdapat beberapa tombol yang dapat mempermudah pengguna untuk melihat gambar pada setiap kotak. Nama dan fungsi setiap tombol dapat dilihat pada Tabel 4.2 Nama dan fungsi tombol tampilan output.

Tabel 4.2 Nama dan fungsi tombol tampilan output

No.	Nama Tombol	Simbol	Fungsi
1	<i>Reset Original View</i>		Mengembalikan halaman utama menjadi tampilan awal
2	<i>Back to Previous View</i>		Mengembalikan tampilan ke satu tampilan sebelum
3	<i>Forward to Next View</i>		Mengubah tampilan ke tampilan setelah
4	<i>Pan / Zoom</i>		Mengaktifkan mode pan and zoom (klik kiri <i>mouse</i> lalu tahan untuk <i>zooming</i> , klik kanan <i>mouse</i> lalu tahan untuk menggeser gambar)
5	<i>Zoom to Rectangle</i>		Memperbesar gambar sesuai area persegi panjang yang digambar
6	<i>Configure Subplot</i>		Menampilkan menu untuk menyesuaikan jarak setiap <i>subplot</i>
7	<i>Save</i>		Menyimpan gambar yang ditampilkan pada halaman utama

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan mengenai skenario uji coba dan evaluasi pada deteksi area hijau pada daerah urab berbasis metode ekstraksi *Oriented FAST And Rotated BRIEF* (ORB) dari citra udara dengan resolusi sangat tinggi. Hasil uji coba didapat dari implementasi pada bab 4 dengan beberapa skenario pengujian yang berbeda. Bab ini berisikan pembahasan mengenai lingkuan pengjian, data pengujian.

5.1. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kakas sebagai berikut:

Tabel 5.1 Spesifikasi lingkungan uji coba

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Intel® Core™ i3-4150 CPU @ 3.50GHz
	Memori	4 GB
Perangkat Lunak	Sistem Operasi	Windows 8 64-bit
	Perangkat Pengembang	IDLE (Python GUI)

5.2. Data Uji Coba

Data yang digunakan untuk uji coba implementasi sistem deteksi area tanaman hijau adalah citra udara perkotaan Sapporo, Jepang pada tahun 2010. Terdapat 50 gambar yang masing-masing memiliki resolusi 1.000 x 1.000 pixel, resolusi spasial 20 cm dan pita warna RGB. Contoh citra masukan yang digunakan sebagai data masukan ditunjukkan oleh Gambar 5.1.



Gambar 5.1 Gambar Uji Coba

5.3. Skenario Uji Coba

Pada subbab ini akan dijelaskan mengenai skenario uji coba yang telah dilakukan. Terdapat beberapa skenario uji coba yang telah dilakukan, diantaranya yaitu:

1. Skenario pengujian 1: Perhitungan nilai recall dan precision menggunakan HSV channel.
2. Skenario pengujian 2: Perhitungan nilai recall dan precision pada proses ekstraksi *feature* ORB

5.4. Skenario Pengujian 1: Perhitungan nilai recall dan precision menggunakan HSV channel

Pada skenario uji coba ini dilakukan uji coba sistem deteksi menggunakan nilai *threshold* untuk menentukan batas nilai kecocokan warna HSV *channel*. Uji coba menggunakan 50 citra udara serta dilakukan perhitungan nilai *recall*, *precision*, dan F1. Nilai tersebut dihitung berdasarkan ketentuan berikut:

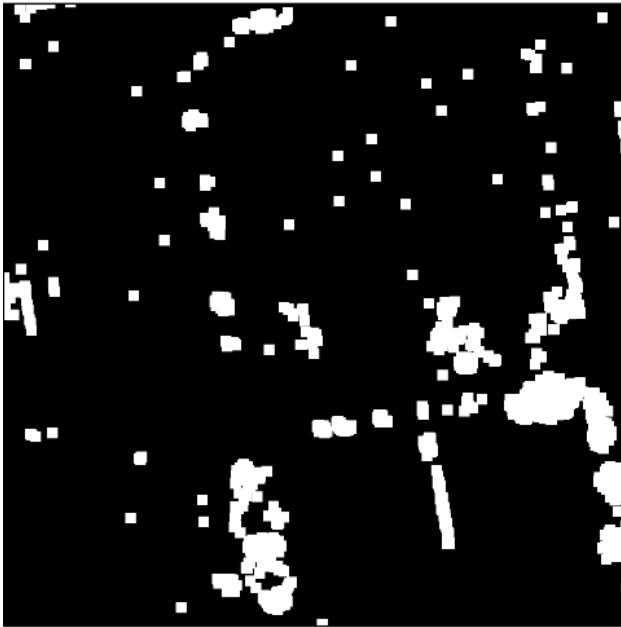
1. *True positive*: penanda dengan objek area tanaman hijau
2. *False negative*: area tanaman hijau namun tanpa penanda
3. *False positive*: penanda area hijau namun bukan objek area tanaman hijau

Untuk memudahkan perhitungan tiga ketentuan tersebut, dilakukan penandaan secara manual dengan menggunakan kaskas bantu coreldraw. Hasil penandaan ini akan digunakan sebagai referensi untuk menghitung kebenaran penanda area tanaman hijau dari setiap proses. Contoh citra hasil penandaan manual dapat dilihat pada Gambar 5.2.



Gambar 5.2 Contoh Citra Hasil Penandaan Manual

Pada skenario ini menggunakan batas nilai HSV *channel* antara (50, 50, 50) sampai (60, 255, 255). Salah satu hasil keluaran dari proses ini dapat dilihat pada Gambar 5.3.



Gambar 5.3 Citra Hasil HSV Channel

Hasil uji coba menggunakan nilai HSV dapat dilihat pada Lampiran A.1 dan Lampiran A.2. Sebagian besar tanaman hijau yang terdeteksi, namun juga mendeteksi objek hijau yang bukan tanaman hijau. Dari 50 citra udara masukan, nilai rata-rata *recall* yang dihasilkan adalah 5,11%. Sedangkan nilai rata-rata *precision* adalah 79,30%. Rata-rata nilai F1 pada proses untuk 50 citra udara adalah 9,45%.

5.5. Skenario Pengujian 2: Perhitungan nilai recall dan presicion pada proses ekstraksi *feature* ORB

Pada skenario ini dilakukan uji coba sistem deteksi area tanaman hijau menggunakan ekstraksi *feature Oriented FAST and Rotated BRIEF* (ORB). Uji coba menggunakan 50 citra udara dan dilakukan perhitungan nilai *recall*, *precision*, dan F1. Nilai tersebut dihitung berdasarkan ketentuan berikut :

1. *True positive*: penanda dengan objek area tanaman hijau
2. *False negative*: area tanaman hijau namun tanpa penanda
3. *False positive*: penanda area hijau namun bukan objek area tanaman hijau

Pada tahap ini, mennggunakan ekstraksi fitur ORB. Hasil dari pengujian ini dapat dilihat pada Gambar 5.4



Gambar 5.4 Hasil Citra Ekstraksi Feature ORB

Hasil uji coba menggunakan ekstraksi *feature Oriented FAST and Rotated BRIEF (ORB)* dapat dilihat pada Lampiran A.3 dan Lampiran A.4. tanaman hijau yang terdeteksi hanya yang terlihat cerah dan tidak mendeteksi tanaman hijau dibalik bayangan gedung, sertatidak mendeteksi objek hijau yang bukan tanaman hijau . Dari 50 citra udara masukan, nilai rata-rata *recall* yang dihasilkan adalah 41,15%. Sedangkan niai rata-rata

precision adalah 49,63%. Rata-rata nilai F1 pada proses untuk 50 citra udara adalah 37,62%.

5.6. Evaluasi

Dari skenario uji coba yang telah digunakan dalam uji coba membe rikan pengaruh terhadap hasil akurasi kinerja implementasi sistem deteksi area tanaman hijau ini. Terdapat beberapa hal yang menjadi evaluasi pada sistem deteksi terdapat objek juga objek lain yang berwarna hijau dan tingkat pencahayaan sangat berpengaruh.

Berdasarkan hasil pengujian pada Lampiran A.1 dan Lampiran A.2 pada sekanrio pengujian pertama dilihat hasil yang dikeluarkan mendeteksi semua warna hijau yang terdapat pada citra. Akan tetapi, untuk citra udara setelah menggunakan ekstraksi *feature* ORB mendekti tanaman hijau yang tidak terkena bayangan gedung, serta tidak mendeteksi objek hijau yang bukan tanaman hijau.

Table 5.1 Nilai rata-rata *precision*, *recalls* dan F1

No.	Uji Skenario	Precissions (%)	Recalls (%)	F1 (%)
1	Nilai HSV	79,30	5,11	9,45
2	Ekstraksi <i>feature</i> ORB	49,63	41,15	37,62

LAMPIRAN

Lampiran A. 1 Hasil uji coba deteksi tanaman hijau menggunakan nilai HSV bagian 1

No.	Nama File	Precissions (%)	Recalls (%)	F1 (%)
1	01_001.png	99.27	5.00	9.51
2	01_002.png	93.66	3.73	7.17
3	01_003.png	88.33	6.25	11.67
4	01_004.png	92.74	4.86	9.23
5	01_005.png	64.39	3.32	6.31
6	01_006.png	89.47	3.44	6.63
7	01_007.png	87.12	5.28	9.96
8	01_008.png	81.74	12.69	21.97
9	01_009.png	87.43	4.00	7.65
10	01_010.png	83.63	7.63	13.99
11	02_001.png	87.85	4.13	7.90
12	02_002.png	91.77	4.35	8.31
13	02_003.png	87.86	3.59	6.91
14	02_004.png	91.99	5.38	10.17
15	02_005.png	84.32	4.45	8.45
16	02_006.png	85.15	2.75	5.34
17	02_007.png	93.57	3.40	6.57
18	02_008.png	83.12	8.21	14.94
19	02_009.png	84.21	6.67	12.36
20	02_010.png	81.95	7.80	14.25
21	03_001.png	60.95	1.96	3.79
22	03_002.png	88.12	2.66	5.16
23	03_003.png	88.26	4.11	7.85
24	03_004.png	91.59	6.85	12.74
25	03_005.png	90.90	4.55	8.67
26	03_006.png	86.34	3.04	5.88
27	03_007.png	75.03	2.66	5.13

**Lampiran A. 2 Hasil uji coba deteksi tanaman hijau
meggunakan nilai HSV bagian 2**

No.	Nama File	Precissions (%)	Recalls (%)	F1 (%)
28	03_008.png	72.98	6.36	11.70
29	03_009.png	67.76	6.27	11.48
30	03_010.png	74.00	6.05	11.19
31	04_001.png	88.91	1.45	2.85
32	04_002.png	82.87	4.36	8.28
33	04_003.png	43.85	6.51	11.33
34	04_004.png	74.48	8.87	15.85
35	04_005.png	70.79	1.08	2.14
36	04_006.png	84.26	2.19	4.27
37	04_007.png	44.62	2.02	3.87
38	04_008.png	71.06	5.69	10.54
39	04_009.png	57.25	11.40	19.02
40	04_010.png	91.78	9.42	17.08
41	05_001.png	65.96	6.92	12.52
42	05_002.png	62.37	2.70	5.18
43	05_003.png	75.08	6.86	12.57
44	05_004.png	63.17	3.54	6.71
45	05_005.png	73.42	2.11	4.10
46	05_006.png	79.65	2.04	3.98
47	05_007.png	84.65	6.23	11.60
48	05_008.png	72.36	8.52	15.24
49	05_009.png	65.33	3.63	6.88
50	05_010.png	77.62	8.57	15.43
	Mean	79.30	5.11	9.45

**Lampiran A. 3 Hasil uji coba deteksi tanaman hijau
menggunakan nilai ekstraksi fitur ORB bagian 1**

No.	Nama File	Precissions (%)	Recalls (%)	F1 (%)
1	01_001.png	97.18	11.32	20.27
2	01_002.png	82.52	19.44	31.47
3	01_003.png	80.95	19.73	31.73
4	01_004.png	75.17	18.89	30.20
5	01_005.png	27.30	39.39	32.25
6	01_006.png	58.41	30.72	40.26
7	01_007.png	46.08	38.23	41.79
8	01_008.png	32.03	77.18	45.27
9	01_009.png	34.76	70.90	46.64
10	01_010.png	67.97	42.99	52.67
11	02_001.png	75.70	16.82	27.52
12	02_002.png	65.66	25.14	36.36
13	02_003.png	58.72	31.44	40.96
14	02_004.png	53.73	46.11	49.63
15	02_005.png	44.37	38.40	41.17
16	02_006.png	58.35	25.91	35.89
17	02_007.png	61.21	38.41	47.20
18	02_008.png	30.17	53.17	38.49
19	02_009.png	49.98	36.43	42.15
20	02_010.png	65.54	39.46	49.26
21	03_001.png	50.06	33.78	40.34
22	03_002.png	60.26	16.73	26.19
23	03_003.png	76.99	18.28	29.55
24	03_004.png	87.32	24.63	38.42
25	03_005.png	77.52	27.49	40.58
26	03_006.png	76.32	22.98	35.33
27	03_007.png	49.14	42.46	45.56

**Lampiran A. 4 Hasil uji coba deteksi tanaman hijau
meggunakan nilai ekstraski fitur ORB bagian 2**

No.	Nama File	Precissions (%)	Recalls (%)	F1 (%)
28	03_008.png	10.46	56.76	17.66
29	03_009.png	36.00	50.20	41.93
30	03_010.png	38.50	35.33	36.85
31	04_001.png	71.63	12.91	21.88
32	04_002.png	47.92	45.87	46.87
33	04_003.png	4.32	75.00	8.18
34	04_004.png	26.72	84.98	40.65
35	04_005.png	33.31	53.06	40.92
36	04_006.png	43.64	39.78	41.62
37	04_007.png	10.56	63.37	18.11
38	04_008.png	38.16	49.83	43.23
39	04_009.png	38.00	72.79	49.94
40	04_010.png	60.76	32.30	42.18
41	05_001.png	26.67	47.22	34.09
42	05_002.png	30.37	44.12	35.98
43	05_003.png	34.60	61.24	44.21
44	05_004.png	20.55	68.86	31.65
45	05_005.png	39.03	35.94	37.42
46	05_006.png	36.72	28.27	31.95
47	05_007.png	60.57	56.65	58.55
48	05_008.png	30.08	68.02	41.71
49	05_009.png	34.23	37.03	35.58
50	05_010.png	65.35	31.74	42.73
	Mean	49.63	41.15	37.62

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan yang diperoleh selama pengerjaan Tugas Akhir dan saran mengenai pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1. Kesimpulan

Dari hasil pengamatan selama proses perancangan, implementasi, dan pengujian perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Metode Thresholding HSV menghasilkan nilai rata-rata *precision* yang dihasilkan lebih tinggi, yaitu sebesar 79.3%, *recall* sebesar 5.11%, dan F1 9.45%.
2. Metode Ekstraksi *Feature ORB* menghasilkan rata-rata *precision* sebesar 49.63%, *recall* yang dihasilkan lebih tinggi, yaitu 41.15% dan F1 37.52%.

6.2. Saran

Berikut merupakan beberapa saran untuk pengembangan sistem di masa yang akan datang. Saran-saran ini didasarkan pada hasil perancangan, implementasi dan pengujian yang telah dilakukan.

1. Perlu adanya preprocessing untuk menghilangkan objek selain tanaman hijau agar dapat menambah nilai hasil deteksi.
2. Untuk meningkatkan akurasi tanaman hijau yang terdeteksi dapat dilakukan dengan menggunakan data *Digital Surface Model* (DSM), berupa nilai ketinggian pohon agar dapat mendeteksi pohon yang terkena bayangan objek lain.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] “Ebiologi - Penyebab penyemaran udara,” [Online]. Available: <http://www.ebiologi.com/2015/07/pencemaran-udara-pengertian-penyebab.html>. [Diakses 20 Juni 2016].
- [2] “Detik - Polusi Udara Indonesia di Posisi 8 Paling Mematikan,” [Online]. Available: <http://news.detik.com/berita/3028851/polusi-udara-indonesia-di-posisi-8-paling-mematikan-ini-kata-menteri-siti>.
- [3] M. Feizas, A. Bardera, J. Rigau, Q. Xu dan M. Sbert, *Information Theory Tools for Image Processing*, Berkeley, California: Morgan & Claypool Publisher series, 2014.
- [4] E. Rublee, V. Rabaud, K. Konolige dan B. Gary, “ORB: an efficient alternative to SIFT or SURF,” dalam *IEEE International conference on Computer Vision*, California, 2011.
- [5] D. G. Viswanathan, “homepages.inf.ed.ac.uk,” [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/AV1FeaturefromAcceleratedSegmentTest.pdf. [Diakses June 2016].
- [6] E. Rosten, R. Porter dan T. Drummond, “Faster and Better: A Machine Learning Approach to Corner Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. %1 dari %232, No. 1, January 2010.
- [7] B. Y. Pratama, “<http://ilmukomputer.org/>,” [Online]. Available: <http://ilmukomputer.org/wp-content/uploads/2014/02/Batra-Operasi-Morfologi-Pada-Citra-Biner.pdf>. [Diakses 20 June 2016].
- [8] V. Christlein, C. Riess, J. Jordan dan C. Riess, “An Evaluation of Popular Copy-Move Forgery Detection

Approaches,” *IEEE Transactions On Information Forensics and Security*, vol. 7, p. 6, 2012.

- [9] B. Y. Pratama, “<http://docplayer.info/>,” [Online]. Available: <http://docplayer.info/278622-Batra-yudha-pratama-m111511006-students-jtk-polban-ac-id.html>. [Diakses 28 May 2016].
- [10] D. N. Rahmah, “digilib.its.ac.id,” [Online]. Available: <http://digilib.its.ac.id/public/ITS-Undergraduate-16849-5107100014-paper-1pdf.pdf>. [Diakses 20 6 2016].

BIODATA PENULIS



Penulis, Azis Arijaya, lahir pada tanggal 25 Juni 1994. Penulis menempuh pendidikan sekolah dasar di SDN Beji 7 Depok. Melanjutkan pendidikan sekolah menengah pertama di SMPN 5 Depok, kemudian pindah ke SMPN 41 Jakarta dan selanjutnya di SMAN 38 Jakarta. Selanjutnya penulis melanjutkan pendidikan sarjana di Jurusan Teknik Informatika, Fakultas Teknologi dan Informasi, Institut Teknologi Sepuluh Nopember Surabaya. Selama kuliah, penulis aktif menjadi asisten mata kuliah Perancangan Perangkat Lunak dan mata kuliah Analisis dan Perancangan Sistem Infomasi di Teknik Informatika dan aktif dalam organisasi tingkat Jurusan sebagai Kepala Departemen Hubungan Luar HMTC, dan tingkat Fakultas sebagai staff *Organization Social Responsibility*. Dalam menyelesaikan pendidikan S1, penulis mengambil bidang minat Komputasi Cerdas dan Visi (KCV) dan memiliki ketertarikan di bidang komputer visi dan juga manajemen SDM. Penulis dapat dihubungi melalui: azis.arijaya@gmail.com.